

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ»**

Факультет физико-математических и естественных наук  
Кафедра прикладной информатики и теории вероятности

**ОТЧЕТ**

**по лабораторной работе**

ТЕМА «Установка и конфигурация операционной системы на виртуальную  
машину»  
по дисциплине «Операционные системы»

Выполнил:

Студент группы НПИбд-02-21

Щербак Маргарита Романовна

«21» апреля 2022г.

Москва 2022

**Цель работы:** приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

### Ход работы:

В первую очередь, скачаем VirtualBox для запуска виртуальных машин с гостевыми операционными системами внутри основной системы. Обычно VM (Virtual Machine) будет отображаться как окно на рабочем столе компьютера, но в зависимости от того, какой из различных интерфейсов VirtualBox используется, она может отображаться в полноэкранном режиме или удаленно на другом компьютере.

Удобнее всего скачивать дистрибутив виртуальной машины с официального сайта [«www.virtualbox.org»](http://www.virtualbox.org), со странички <https://www.virtualbox.org/wiki/Downloads> (см. рис 1.1).

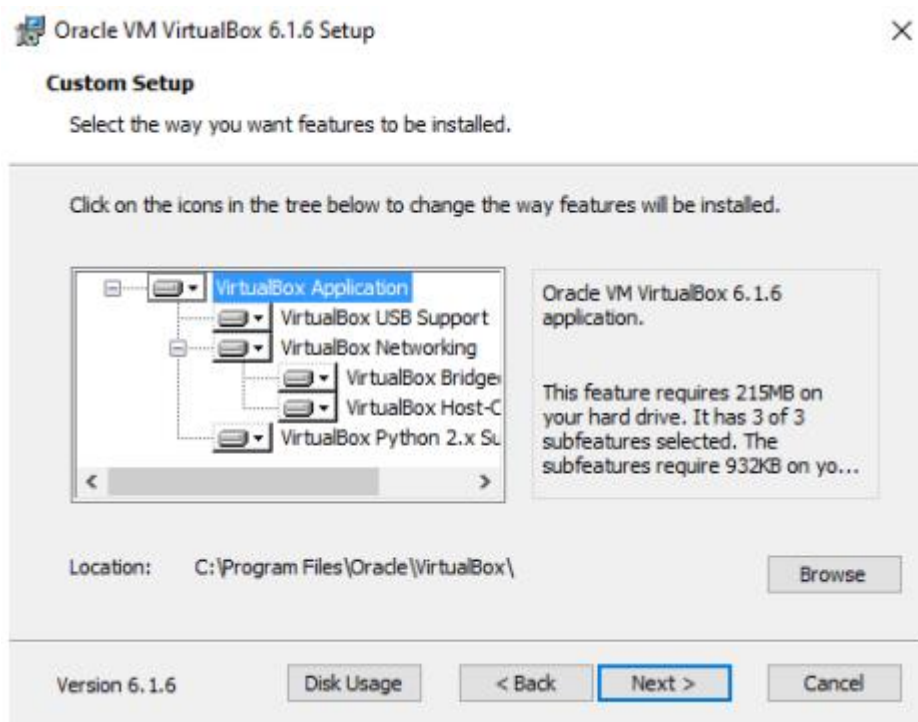


(рис. 1.1)

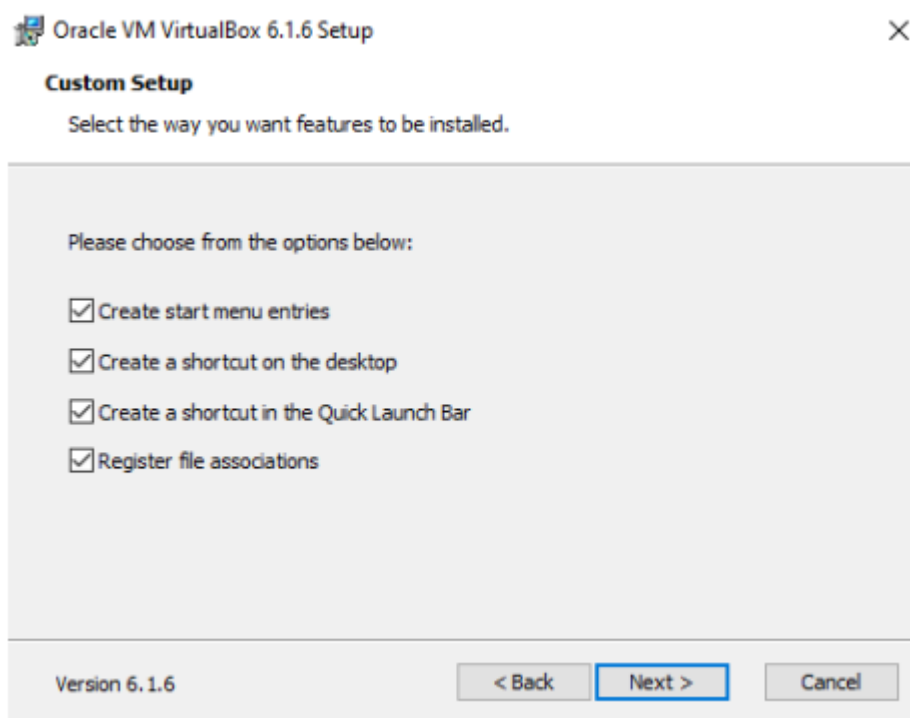
Теперь перейдем к установке VirtualBox и его настройке. Нажимаем по установщику в формате .exe для систем на базе Windows и нажимаем каждый раз кнопку «Next» (см. рис. 2.1 - 2.4). На рисунке 2.5 кнопку «Install», а на рисунке 2.6 – «Finish».



(рис. 2.1)



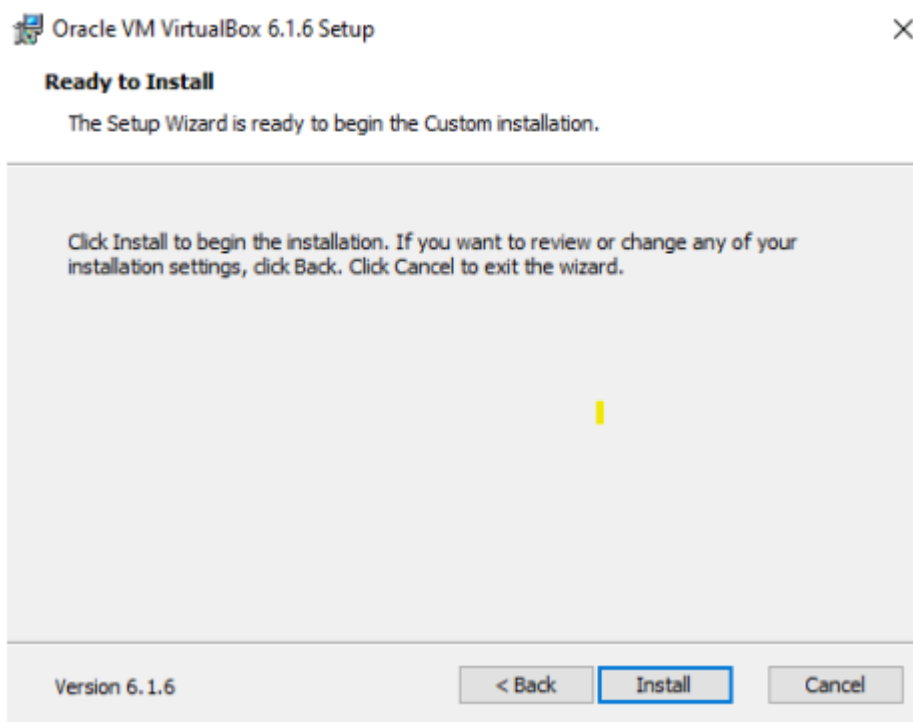
(рис. 2.2)



(рис. 2.3)



(рис. 2.4)



(рис. 2.5)



(рис. 2.6)

Скачала файл Fedora 35 (Рис.3)

Создала новую виртуальную машину. В VirtualBoxвыберла Машина Создать. Указала имя виртуальной машины mrshcherbak (Рис.1), тип операционной системы — Linux, Fedora(Рис.1). Указала размер основной памяти виртуальной машины — от 2048 МБ(Рис.2). Задала конфигурацию жёсткого

диска — загрузочный, VDI (VirtualBox Disk Image), динамический виртуальный диск.

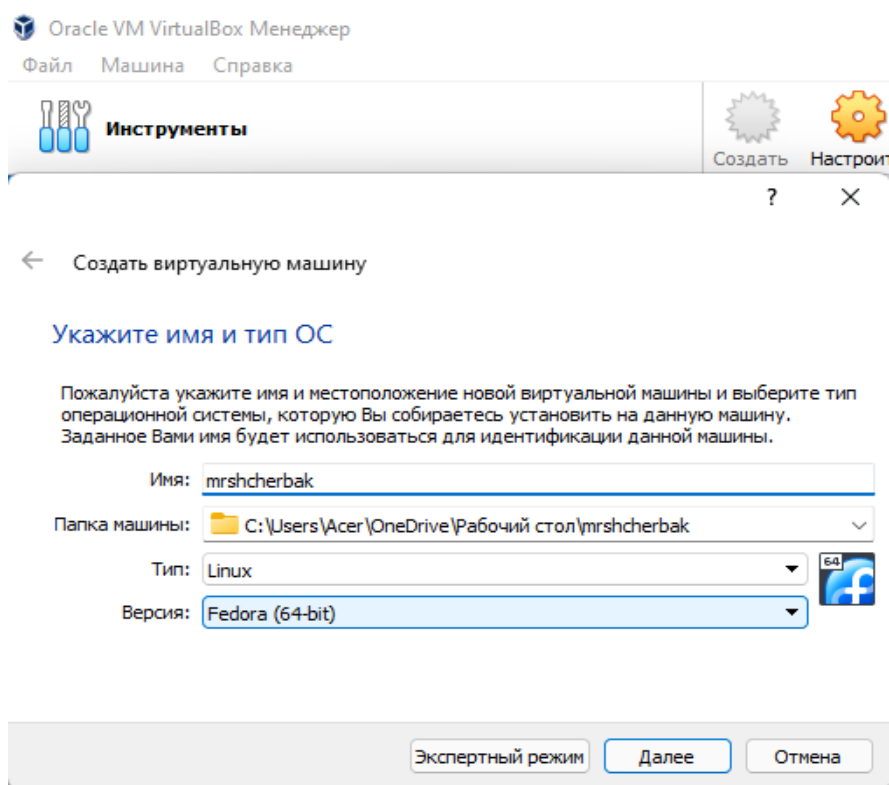


Рис.1

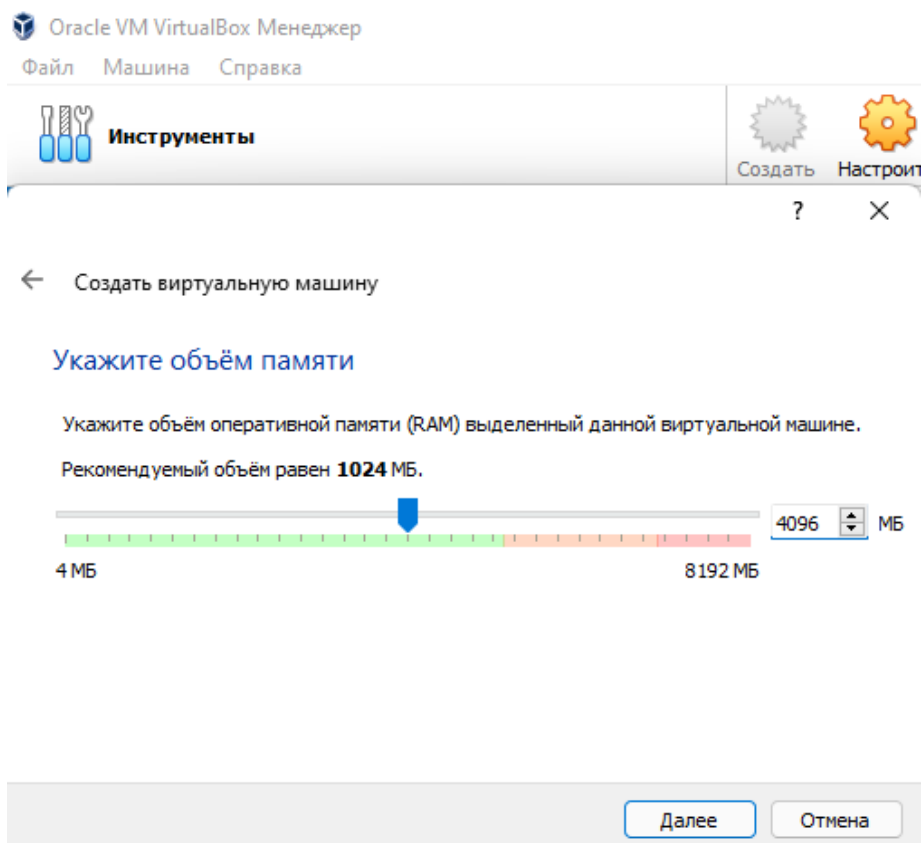
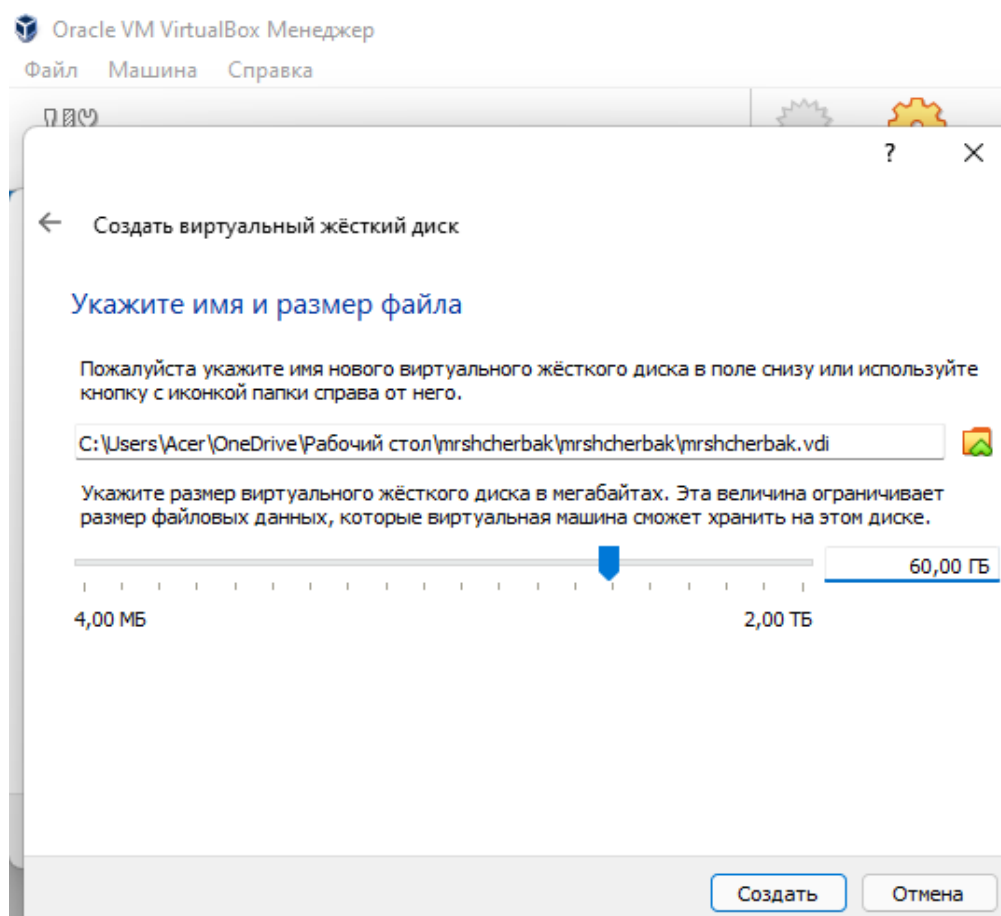


Рис.2

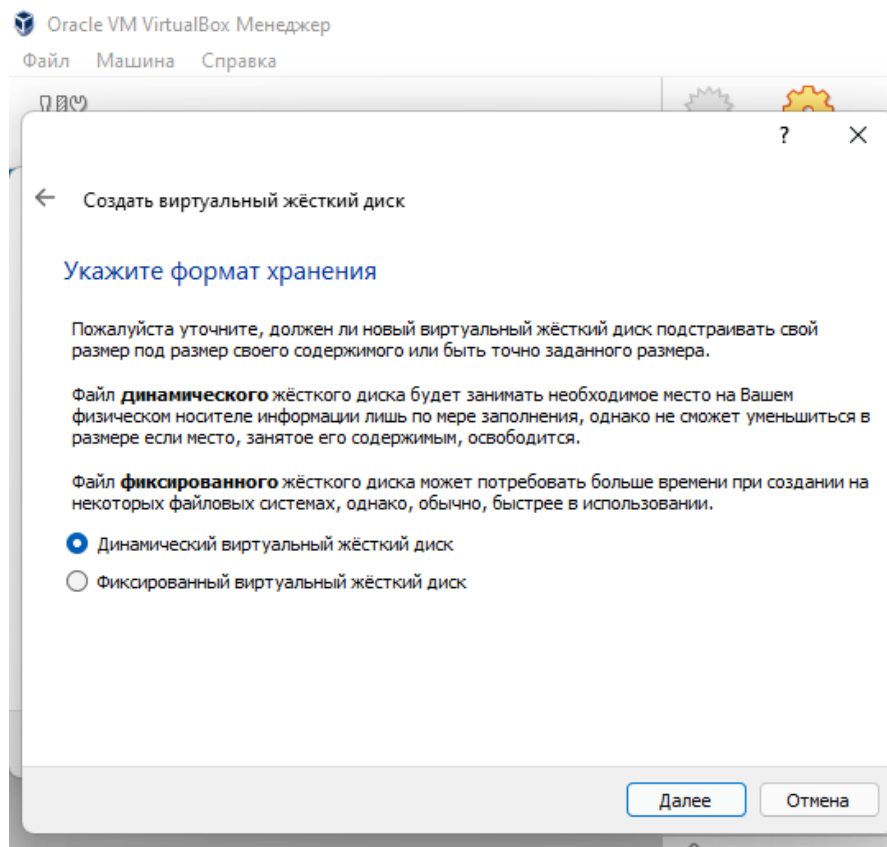
Указываю имя и размер файла:

Размер виртуального диска устанавливаем по желанию, но не меньше, чем указано в требованиях операционной системы. Жмем «Создать». Программа VirtualBox сама рекомендует объем жесткого диска для конкретной операционной системы. Однако стоит выбирать больший объем памяти для установки утилит и дополнительных программ, а также для хранения документов и файлов.



Задаю конфигурацию жёсткого диска:

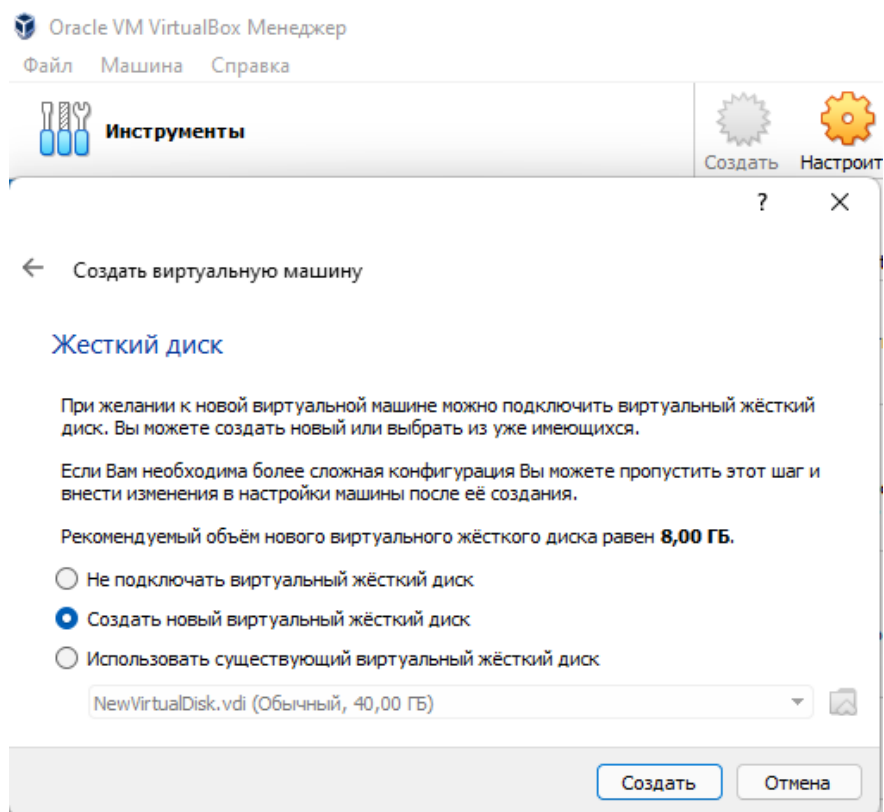
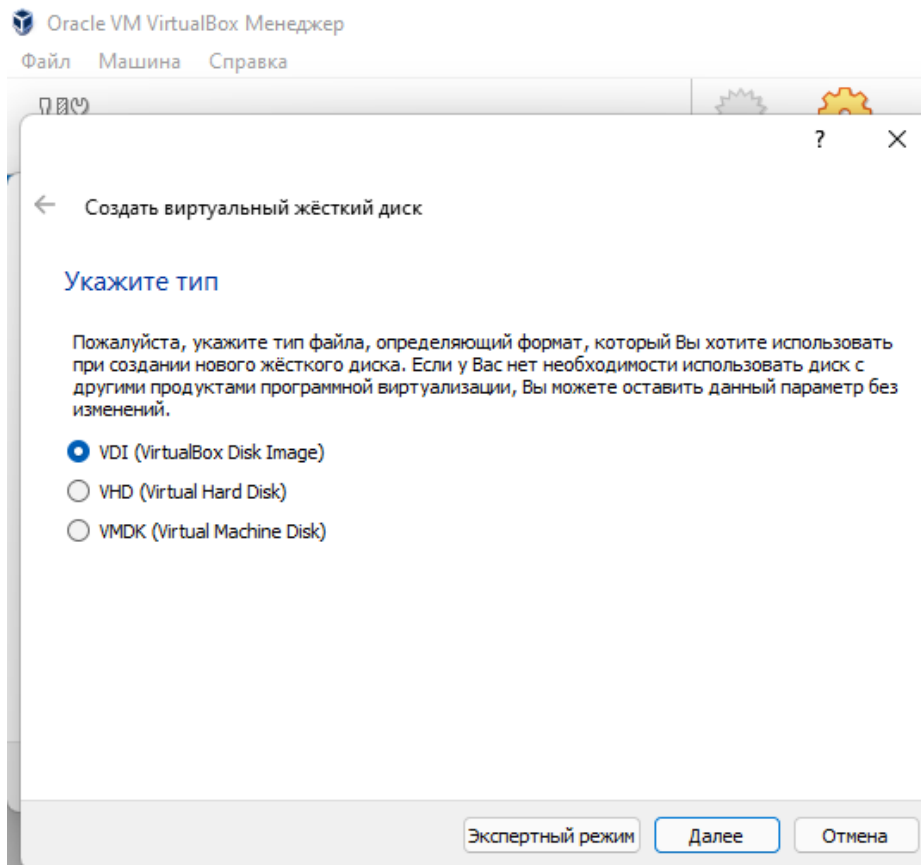
Файл динамического жесткого диска будет занимать небольшое место на физическом жестком диске вашего компьютера. По мере заполнения данными он будет увеличиваться в размере до предельного объема. Файл фиксированного жесткого диска сразу займет весь объем виртуального жесткого диска. Что касается формата хранения, лучше выбрать «Динамический», так это сэкономит время и место на физическом диске, потеряв, правда, немного в производительности.



В следующем окне необходимо будет выбрать тип виртуального жесткого диска. Вы можете выбрать следующие форматы дисков (виртуальных контейнеров):

- VDI (VirtualBox Disk Image) — формат диска VirtualBox.
- VMDK (Virtual Machine Disk) — формат диска VMware.
- VHD (Virtual Hard Disk) — формат диска Microsoft. Если не планируется использовать создаваемый виртуальный накопитель с другими инструментами программной виртуализации, стоит оставить предлагаемый по умолчанию тип жесткого диска «VDI (VirtualBox Disk Image)»





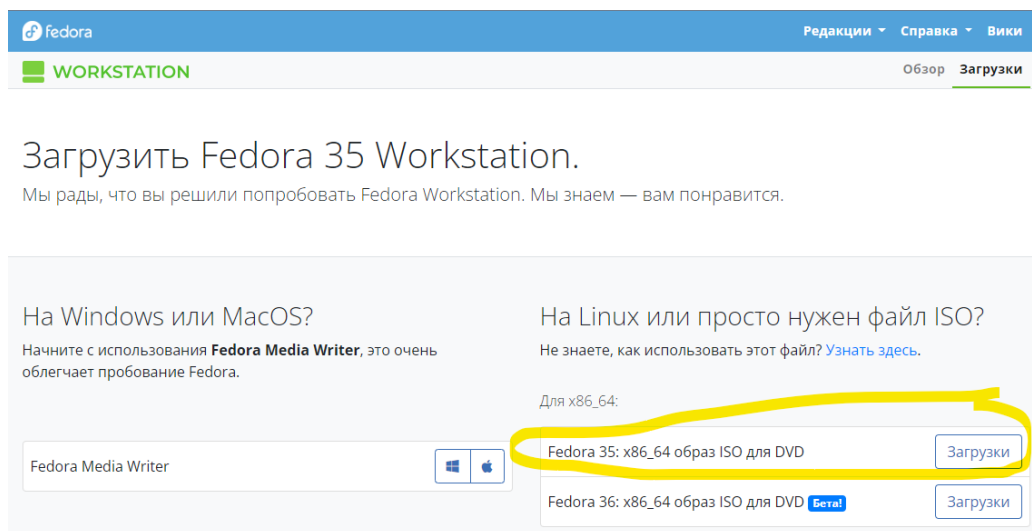
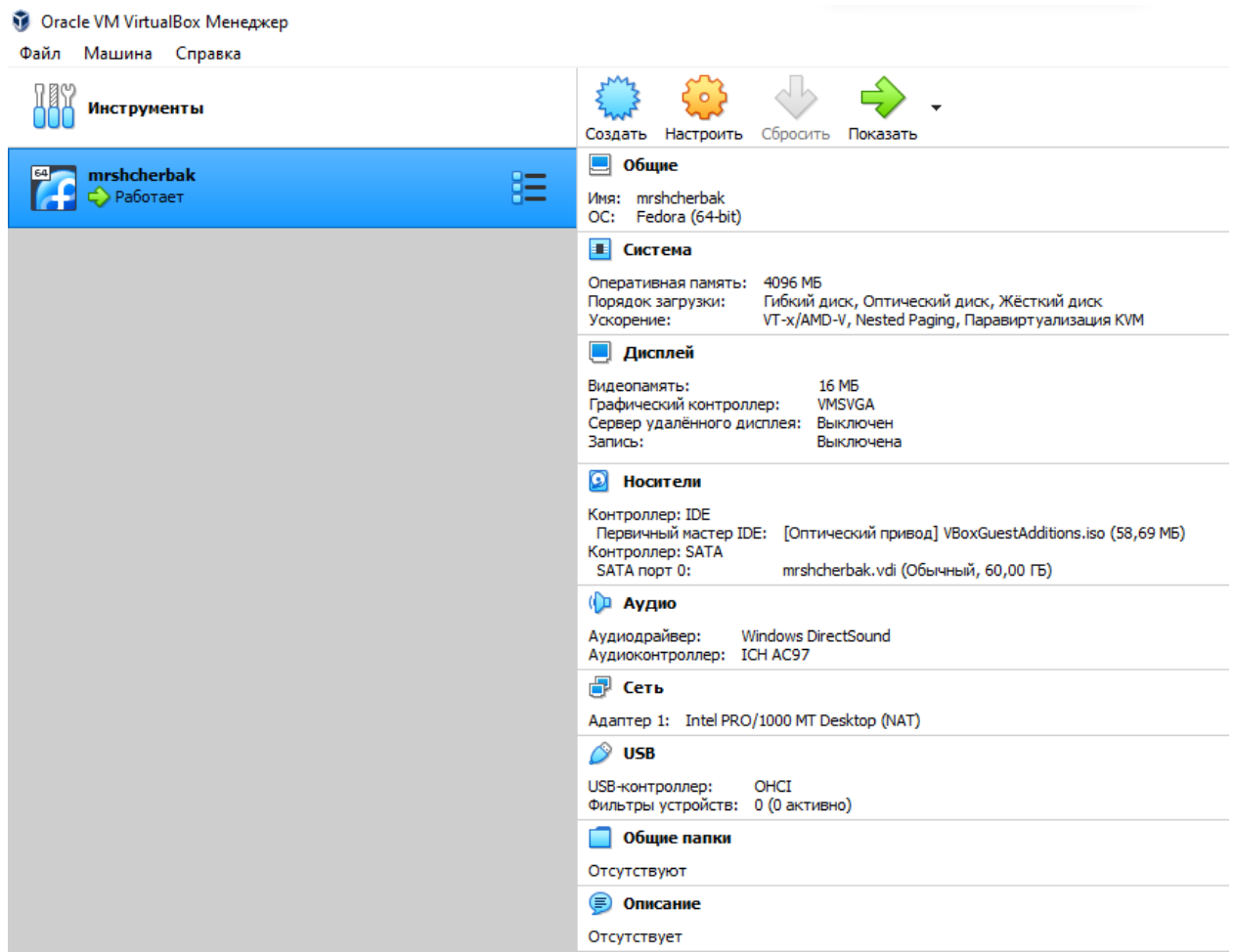


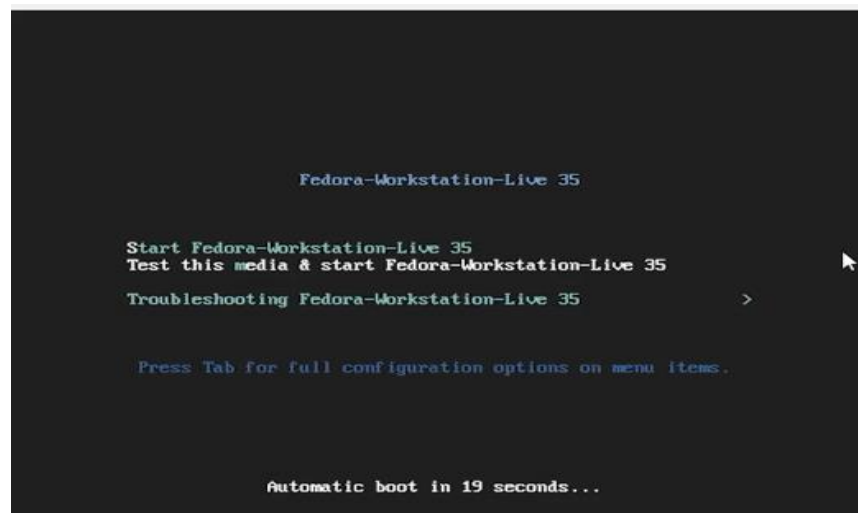
Рис.3

Далее мы настраиваем наш файл .iso Добавили в пункте «Носители» этот файл Fedora.

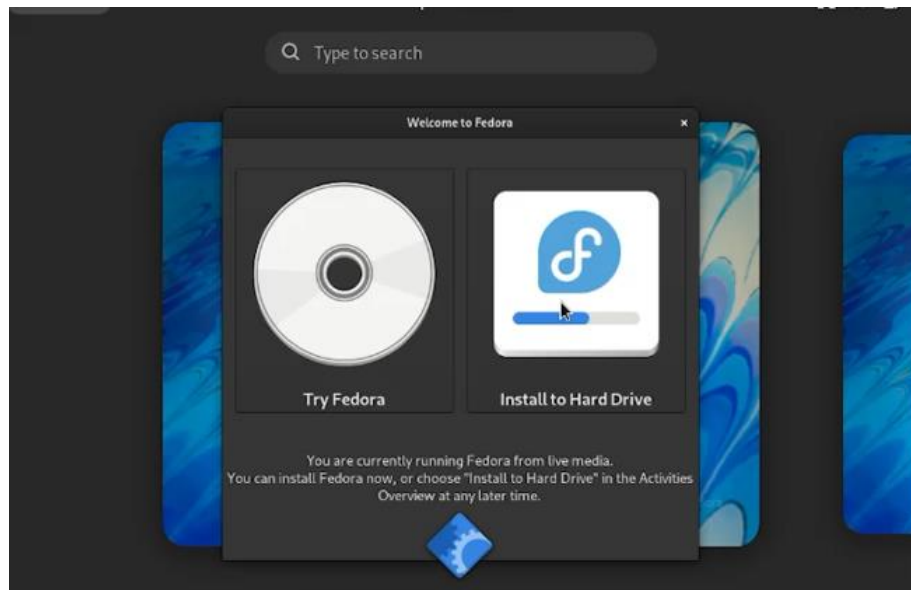
Категория «Носители» в настройках виртуальной машины позволяет подключать к виртуальной машине виртуальный жесткий диск, CD/DVD, дискету и диски.



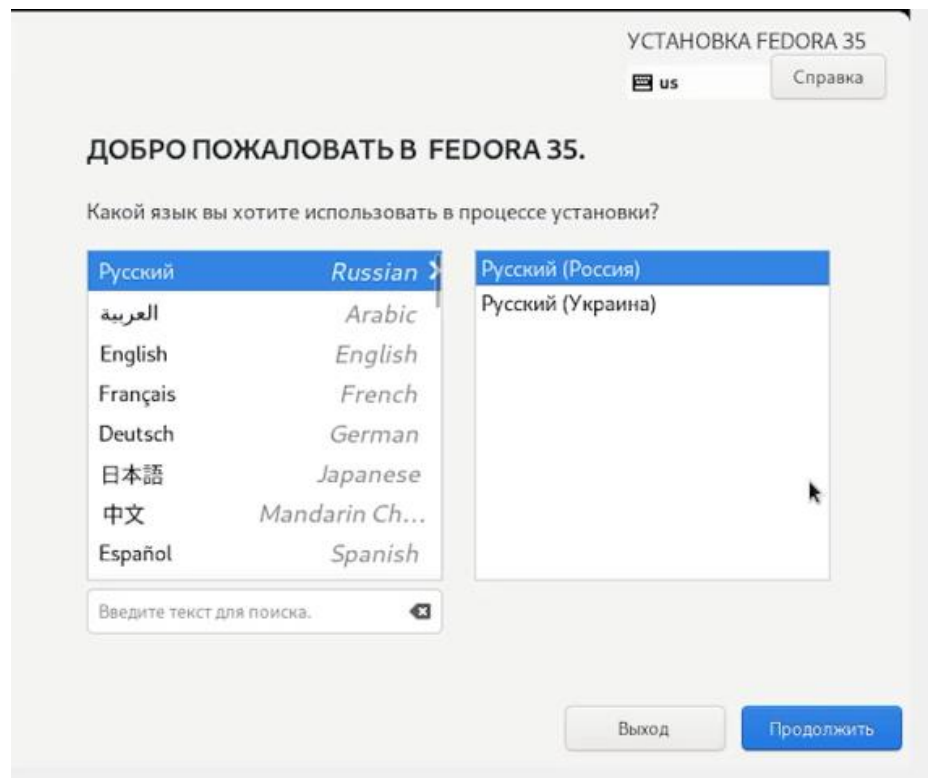
Запускаем виртуальную машину. Ждём установки и загрузки.

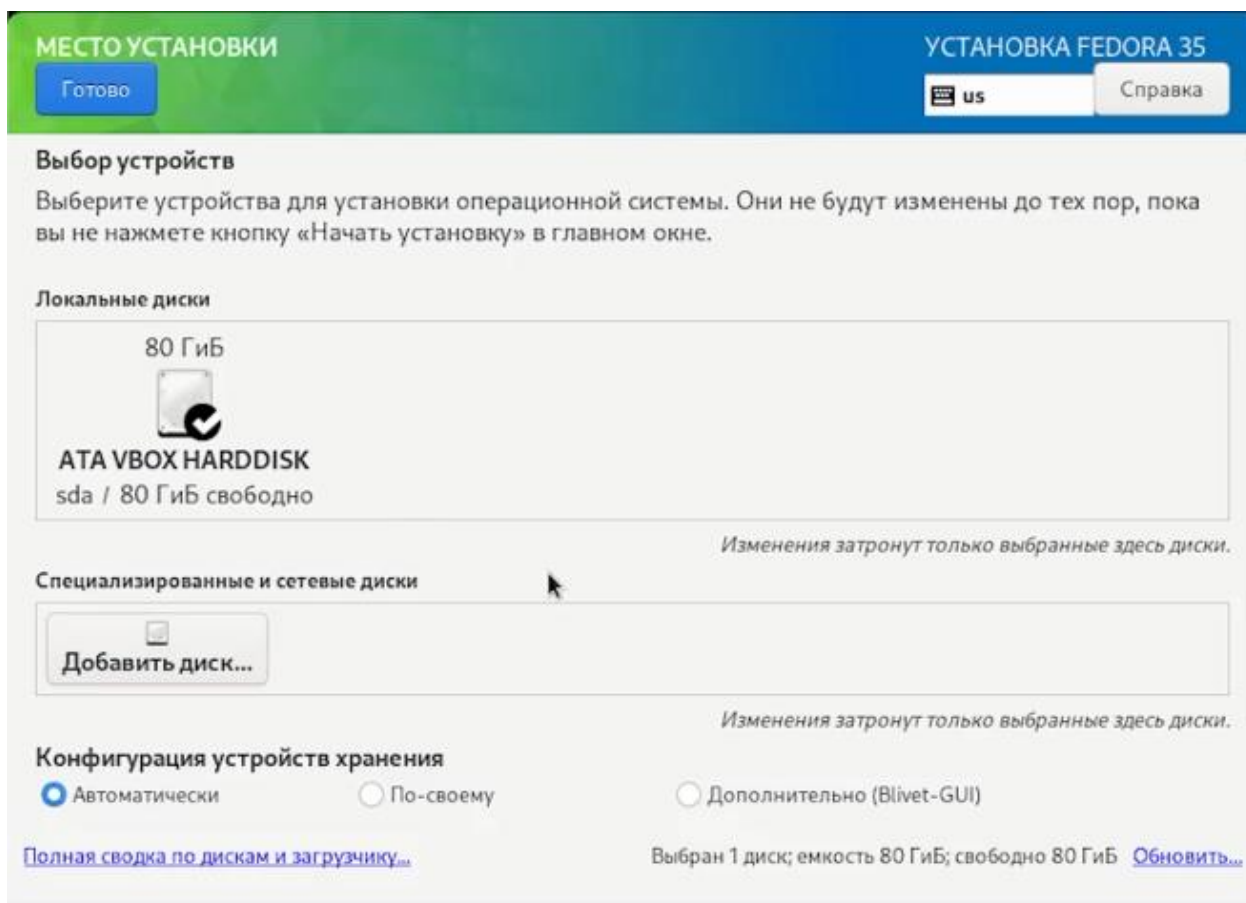
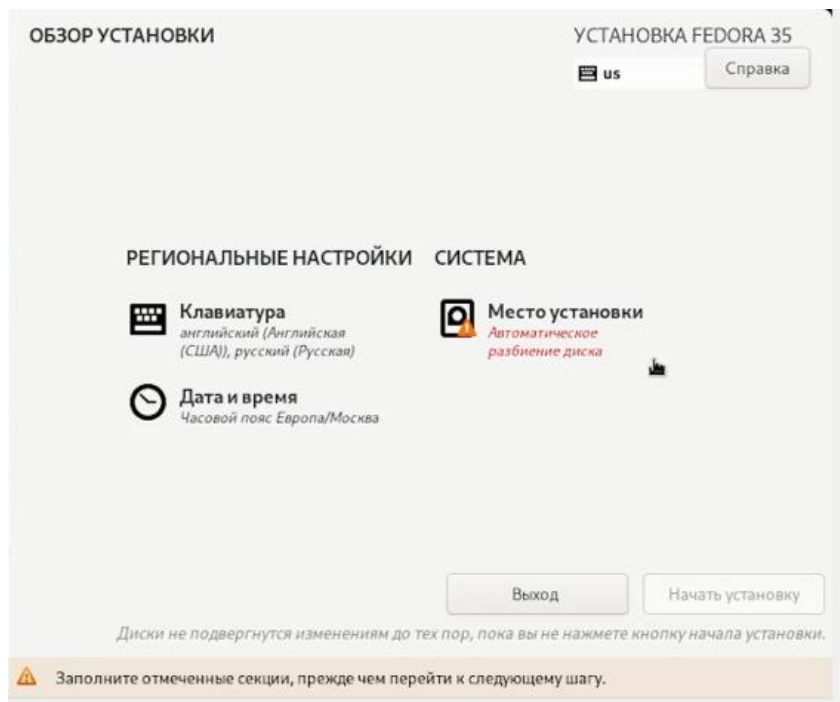


В всплывающем окне выбираем «InstalltoHarddrive»

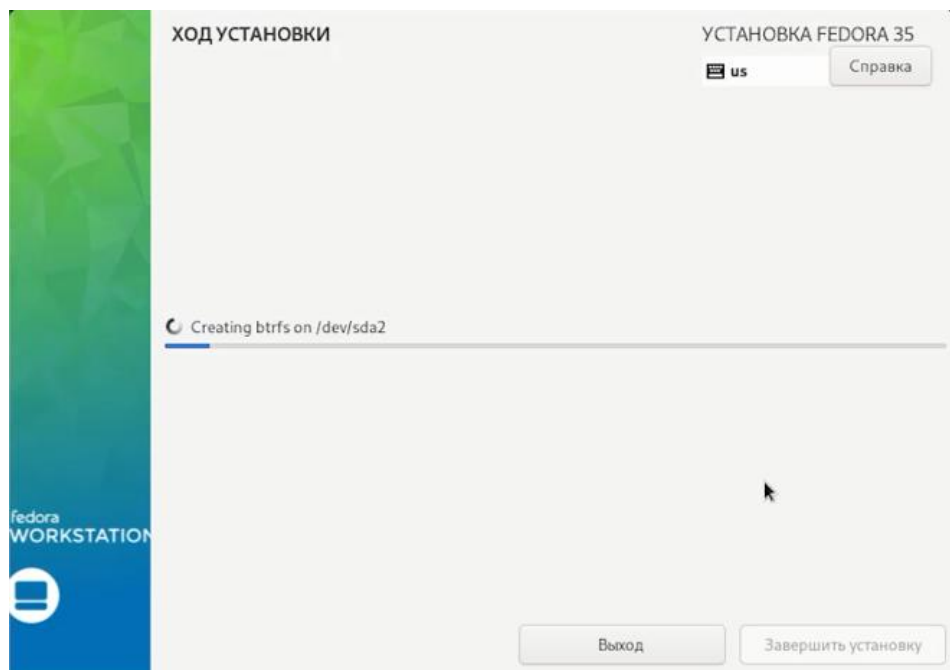


Через какое-то время в появившемся окне выбираем нужный язык(русский), выбираем необходимый часовой пояс, место установки.

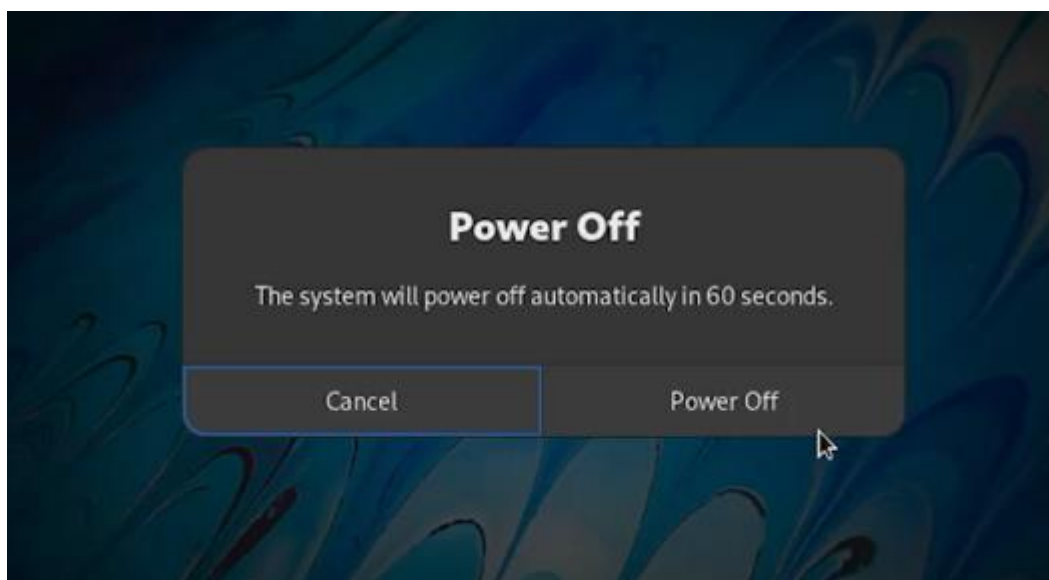


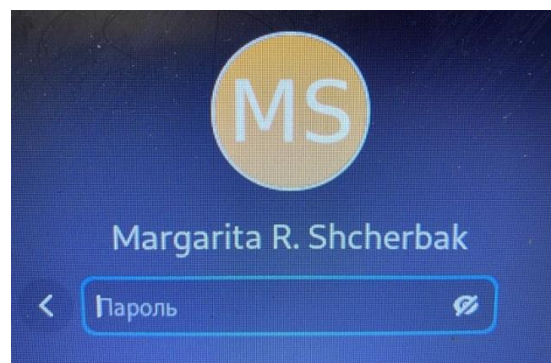
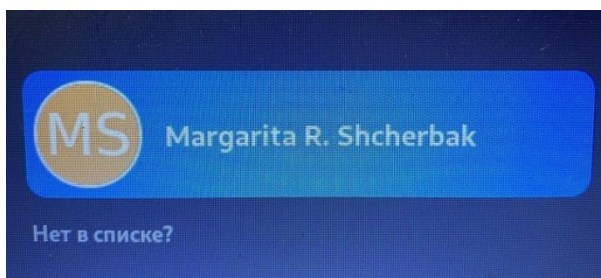


Дальше следует ход установки, после этого выбираем «завершить установку».



Выполняем выключение виртуальной машины, потом изъятие диска из привода, а после вновь запускаем ее, вводим свое имя, создаем пароль





### Домашнее задание:

В окне терминала проанализировала последовательность загрузки системы, выполнив команду `dmesg`.

```
mrshcherbak@fedora:~$ dmesg
[ 0.000000] Linux version 5.14.10-300.fc35.x86_64 (mockbuild@bkernel01.iad2.fedoraproject.org) (gcc (GCC) 11.2.1 20210728 (Red Hat 11.2.1-1), GNU ld version 2.37-10.fc35) #1 SMP Thu Oct 7 20:48:44 UTC 2021
[ 0.000000] Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-5.14.10-300.fc35.x86_64 root=UUID=da95438b-b1d1-441d-a5ec-58a02e5d3720 ro rootflags=subvol=root rhgb quiet
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[ 0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[ 0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.
[ 0.000000] signal: max sigframe size: 1776
[ 0.000000] BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x0000000000009fbff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000009fc00-0x0000000000009ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x000000000000f0000-0x000000000000fffff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000000100000-0x000000000000dffff] usable
[ 0.000000] BIOS-e820: [mem 0x000000000dfff0000-0x000000000dfffffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
```

Получили следующую информацию.

1. Версия ядра Linux (Linux version):

Для того, чтобы найти информацию о версии ОС, я использовала команду `dmesg | grep -i "Linux version"`.

Linux version 5.14.10-300.fc35.x86\_64



```
mrshcherbak@fedora:~$ dmesg | grep -i "Linux version"
[ 0.000000] Linux version 5.14.10-300.fc35.x86_64 (mockbuild@bkernel01.iad2.fedoraproject.org) (gcc (GCC) 11.2.1 20210728 (Red Hat 11.2.1-1), GNU ld version 2.37-10.fc35) #1 SMP Thu Oct 7 20:48:44 UTC 2021
mrshcherbak@fedora ~$
```

## 2. Частота процессора (Detected Mhz processor) 207.998 МГц

```
mrshcherbak@fedora ~$ dmesg | grep -i "processor"
[ 0.000008] tsc: Detected 2207.998 MHz processor
[ 0.240696] smpboot: Total of 1 processors activated (4415.99 BogoMIPS)
[ 0.441074] ACPI: Added _OSI(Processor Device)
[ 0.441077] ACPI: Added _OSI(Processor Aggregator Device)
mrshcherbak@fedora ~$
```

## 3. Модель процессора (CPU0)

Команда `dmesg | grep -i "CPU0"` вывела информацию о модели процессора.

```
mrshcherbak@fedora ~$ dmesg | grep -i "CPU0"
[ 0.239939] smpboot: CPU0: Intel(R) Core(TM) i3-8130U CPU @ 2.20GHz (family: 0x6, model: 0x8e, stepping: 0xa)
mrshcherbak@fedora ~$
```

## 4. Для того, чтобы посмотреть объем доступной оперативной памяти, я использовал команду `dmesg | grep -i "Memory"`.

```
[ 0.239939] smpboot: CPU0: Intel(R) Core(TM) i3-8130U CPU @ 2.20GHz (family: 0x6, model: 0x8e, stepping: 0xa)
mrshcherbak@fedora ~$ dmesg | grep -i "Memory available"
mrshcherbak@fedora ~$ dmesg | grep -i "Memory"
[ 0.001901] ACPI: Reserving FACP table memory at [mem 0xdfff00f0-0xdfff01e3]
[ 0.001903] ACPI: Reserving DSDT table memory at [mem 0xdfff0470-0xdfff2794]
[ 0.001904] ACPI: Reserving FACS table memory at [mem 0xdfff0200-0xdfff023f]
[ 0.001905] ACPI: Reserving FACS table memory at [mem 0xdfff0200-0xdfff023f]
[ 0.001906] ACPI: Reserving APIC table memory at [mem 0xdfff0240-0xdfff0293]
[ 0.001907] ACPI: Reserving SSDT table memory at [mem 0xdfff02a0-0xdfff046b]
[ 0.019416] Early memory node ranges
[ 0.031532] PM: hibernation: Registered nosave memory: [mem 0x00000000-0x00000fff]
[ 0.031536] PM: hibernation: Registered nosave memory: [mem 0x0009f000-0x0009ffff]
[ 0.031538] PM: hibernation: Registered nosave memory: [mem 0x000a0000-0x000effff]
[ 0.031540] PM: hibernation: Registered nosave memory: [mem 0x000f0000-0x000fffff]
[ 0.031543] PM: hibernation: Registered nosave memory: [mem 0xdfff0000-0xdfffffff]
[ 0.031544] PM: hibernation: Registered nosave memory: [mem 0xe0000000-0xfefbffff]
[ 0.031546] PM: hibernation: Registered nosave memory: [mem 0xfec00000-0xfec0ffff]
[ 0.031548] PM: hibernation: Registered nosave memory: [mem 0xfec01000-0xfedfffff]
[ 0.031549] PM: hibernation: Registered nosave memory: [mem 0xfef00000-0xfef0ffff]
[ 0.031551] PM: hibernation: Registered nosave memory: [mem 0xfef01000-0xffffbfff]
[ 0.031552] PM: hibernation: Registered nosave memory: [mem 0xffffc000-0xffffffff]
[ 0.067141] Memory: 3967200K/4193848K available (16393K kernel code, 3531K rdata, 10388K rodata, 2872K init, 4908K bss, 226388K reserved, 0K cma-reserved)
[ 0.137542] Freeing SMP alternatives memory: 44K
[ 0.240998] x86/mm: Memory block size: 128MB
[ 0.691950] Non-volatile memory driver v1.3
[ 1.522684] Freeing initrd memory: 31928K
[ 1.599354] Freeing unused decrypted memory: 2036K
[ 1.600639] Freeing unused kernel image (initmem) memory: 2872K
[ 1.602934] Freeing unused kernel image (text/rodata gap) memory: 2036K
[ 1.603453] Freeing unused kernel image (rodata/data gap) memory: 1900K
[ 4.502648] [TTM] Zone kernel: Available graphics memory: 2004138 KiB
[ 4.502817] [drm] Max dedicated hypervisor surface memory is 507904 KiB
[ 4.502818] [drm] Maximum display memory size is 16384 KiB
mrshcherbak@fedora ~$
```



```
mrshcherbak@fedora:~
[ 0.031546] PM: hibernation: Registered nosave memory: [mem 0xfec00000-0xfec00fff]
[ 0.031548] PM: hibernation: Registered nosave memory: [mem 0xfec01000-0xfedffff]
[ 0.031549] PM: hibernation: Registered nosave memory: [mem 0xfec00000-0xfec00fff]
[ 0.031551] PM: hibernation: Registered nosave memory: [mem 0xfec01000-0xfecffff]
[ 0.031552] PM: hibernation: Registered nosave memory: [mem 0xfec00000-0xfecffff]
[ 0.067141] Memory: 3967200K/4193848K available (16393K kernel code, 3531K rw data, 10388K rodata, 2872K init, 4908K bss, 226388K reserved, 0K cma-reserved)
[ 0.137542] Freeing SMP alternatives memory: 44K
[ 0.240998] x86/mm: Memory block size: 128MB
[ 0.691950] Non-volatile memory driver v1.3
[ 1.522684] Freeing initrd memory: 31928K
[ 1.599354] Freeing unused decrypted memory: 2036K
[ 1.600639] Freeing unused kernel image (initmem) memory: 2872K
[ 1.602934] Freeing unused kernel image (text/rodata gap) memory: 2036K
[ 1.603453] Freeing unused kernel image (rodata/data gap) memory: 1900K
[ 4.502648] [TTM] Zone kernel: Available graphics memory: 2004138 KiB
[ 4.502817] [drm] Max dedicated hypervisor surface memory is 507904 KiB
[ 4.502818] [drm] Maximum display memory size is 16384 KiB
[mrshcherbak@fedora ~]$ b
```

##### 5. Тип обнаруженного гипервизора (Hypervisor detected).

Гипервизор - программа или аппаратная схема, обеспечивающая или позволяющая одновременное, параллельное выполнение нескольких операционных систем на одном и том же хост-компьютере. Гипервизор также обеспечивает изоляцию операционных систем друг от друга, защиту и безопасность, разделение ресурсов между различными запущенными ОС и управление ресурсами.

У меня это KVM (Kernel-based Virtual Machine)

```
[mrshcherbak@fedora ~]$ dmesg | grep -i "Hypervisor detected"
[ 0.000000] Hypervisor detected: KVM
[mrshcherbak@fedora ~]$ dmesg | grep -i "Hypervisor"
[ 0.000000] Hypervisor detected: KVM
[ 0.123499] SRBDS: Unknown: Dependent on hypervisor status
[ 4.502817] [drm] Max dedicated hypervisor surface memory is 507904 KiB
[mrshcherbak@fedora ~]$
```

KVM – гипервизор, созданный в октябре 2006 года, был почти сразу интегрирован с основной веткой ядра Linux версии 2.6.20., выпущенной в начале 2007 года. Позже KVM был адаптирован как модуль ядра в FreeBSD. В KVM включены загружаемый модуль ядра `kvm.ko`, отвечающий за виртуализацию, процессорно-специфический загружаемый модуль для AMD или Intel `kvm-amd.ko` либо `kvm-intel.ko`, и компоненты пользовательского режима QEMU. KVM – полностью открытое ПО по лицензии GNU GPL и GNU LGPL.

6. Для получения информации о типе файловой системы корневого раздела и последовательности монтирования файловых систем я использовал команду `dmesg | grep -i "mount"`.

у меня тип файловой системы ext4.

```
mrshcherbak@fedora:~  
[ 0.562568] smpboot: Total of 1 processors activated (4415.99 BogoMIPS)  
[ 0.876359] ACPI: Added _OSI(Processor Device)  
[ 0.876363] ACPI: Added _OSI(Processor Aggregator Device)  
[mrshcherbak@fedora ~]$ dmesg | grep -i "mount"  
[ 0.431235] Mount-cache hash table entries: 8192 (order: 4, 65536 bytes, linear)  
[ 0.431256] Mountpoint-cache hash table entries: 8192 (order: 4, 65536 bytes, linear)  
[ 9.884138] systemd[1]: Set up automount Arbitrary Executable File Formats File System Automount Point.  
[ 9.933401] systemd[1]: Mounting Huge Pages File System...  
[ 9.943650] systemd[1]: Mounting POSIX Message Queue File System...  
[ 9.956611] systemd[1]: Mounting Kernel Debug File System...  
[ 9.981813] systemd[1]: Mounting Kernel Trace File System...  
[ 10.184577] systemd[1]: Starting Remount Root and Kernel File Systems...  
[ 10.292987] systemd[1]: Mounted Huge Pages File System.  
[ 10.300586] systemd[1]: Mounted POSIX Message Queue File System.  
[ 10.303569] systemd[1]: Mounted Kernel Debug File System.  
[ 10.308330] systemd[1]: Mounted Kernel Trace File System.  
[ 10.392172] systemd[1]: Mounting FUSE Control File System...  
[ 10.420164] systemd[1]: Mounting Kernel Configuration File System...  
[ 14.754849] EXT4-fs (sda1): mounted filesystem with ordered data mode. Opts: (null). Quota mode: none.  
[mrshcherbak@fedora ~]$
```

```
[mrshcherbak@fedora ~]$ df -Th  
Файловая система Тип Размер Использовано Дост Использовано% Смонтировано в  
devtmpfs devtmpfs 1,9G 0 1,9G 0% /dev  
tmpfs tmpfs 2,0G 0 2,0G 0% /dev/shm  
tmpfs tmpfs 783M 1,4M 782M 1% /run  
/dev/sda2 btrfs 59G 3,4G 55G 6% /  
tmpfs tmpfs 2,0G 80K 2,0G 1% /tmp  
/dev/sda2 btrfs 59G 3,4G 55G 6% /home  
/dev/sda1 ext4 974M 172M 736M 19% /boot  
tmpfs tmpfs 392M 124K 392M 1% /run/user/1000  
/dev/sr0 iso9660 59M 59M 0 100% /run/media/mrshcherbak/VBox_GAs_6.1.34  
[mrshcherbak@fedora ~]$
```

7. Последовательность монтирования файловых систем.

для построчного просмотра последовательность загрузки системы можно использовать команду `sudo dmesg | less`.



```
mrshcherbak@fedora:~ — less
[ 0.000000] Linux version 5.14.10-300.fc35.x86_64 (mockbuild@bkernel01.iad2.fedoraproject.org) (gcc (GCC) 11.2.1 20210728 (Red Hat 11.2.1-1), GNU ld version 2.37-10.fc35) #1 SMP Thu Oct 7 20:48:44 UTC 2021
[ 0.000000] Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-5.14.10-300.fc35.x86_64 root=UUID=da95438b-b1d1-441d-a5ec-58a02e5d3720 ro rootflags=subvol=root rhgb quiet
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[ 0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[ 0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.
[ 0.000000] signal: max sigframe size: 1776
[ 0.000000] BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000009fbff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000000009fc00-0x00000000000009ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000000f0000-0x0000000000000fffff] reserved
[ 0.000000] BIOS-e820: [mem 0x000000000000100000-0x000000000000dfffff] usable
[ 0.000000] BIOS-e820: [mem 0x000000000000dfff0000-0x000000000000dfffffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
:

[mrshcherbak@fedora ~]$ dmesg | grep -i "mounted"
[ 7.820198] systemd[1]: Mounted Huge Pages File System.
[ 7.822752] systemd[1]: Mounted POSIX Message Queue File System.
[ 7.826344] systemd[1]: Mounted Kernel Debug File System.
[ 7.826728] systemd[1]: Mounted Kernel Trace File System.
[ 11.249817] EXT4-fs (sda1): mounted filesystem with ordered data mode. Opts: (null). Quota mode: none.
[mrshcherbak@fedora ~]$
```

## Ответы на контрольные вопросы:

1. Когда пользователь регистрируется в системе (проходит процедуру авторизации, например, вводя системное имя и пароль), он идентифицируется с учётной записью, в которой система хранит информацию о каждом пользователе: его системное имя и некоторые другие сведения, необходимые для работы с ним. Именно с учётными записями, а не с самими пользователями, и работает система. Таким образом, учетная запись пользователя содержит:

- 1) Системное имя (user name)
- 2) Идентификатор пользователя (UID)
- 3) Идентификатор группы (GID)
- 4) Полное имя (full name)

5) Домашний каталог (home directory)

6) Начальная оболочка (login shell)

## 2. Команды терминала:

1) `man [аргумент]` – для получения справки по команде. Пример: `man ls` – выведет информацию о команде `ls`;

2) `cd [путь]` – для перемещения по файловой системе. Пример: `cd /` – для перехода в корневой раздел;

3) `ls [опции]` – для просмотра содержимого каталога. Пример: `ls -alS /` – выведет подробной информации о файлах в корневом каталоге с сортировкой по алфавиту;

4) `du [опции] [путь]` – для определения объёма каталога. Пример: `du -h ~/”Изображения”` – выведет размер каталога «Изображения» с указанием единицы измерения;

5) `mkdir [опции] [путь]` / `rmdir [опции] [путь]` / `rm [опции] [путь]` – для создания / удаления каталогов / файлов. Примеры: `mkdir -pv ~/MyDir ~/MyDir2` – создаст каталоги `MyDir` и `MyDir2` в домашней директории; `rmdir -v ~/MyDir` – удалит каталог `MyDir` из домашней директории; `rm -rv ~/MyDir2` – удалит каталог `MyDir2` из домашней директории;

6) `chmod [аргументы] [путь]` - для задания определённых прав на файл / каталог. Пример: `chmod o-w ~/Readme.txt` – отнимет право изменения текстового файла `Readme.txt` остальным пользователям, кроме владельца и группы владельца.

7) `history [опции]` – для просмотра истории команд. Пример: `history -c` – очистит историю команд Linux.

**3. Файловая система** - порядок, определяющий способ организации, хранения и именования данных на носителях информации в компьютерах, а также в другом электронном оборудовании.

Файловые системы в Linux используются не только для работы с файлами на диске, но и для хранения данных в оперативной памяти или доступа к конфигурации ядра во время работы системы.

Каждый дистрибутив Linux позволяет использовать одну из этих файловых систем, каждая из них имеет свои преимущества и недостатки:

- Ext2;
- Ext3;

- Ext4;
- JFS;
- ReiserFS;
- XFS;
- Btrfs;
- ZFS;

Все они включены в ядро и могут использоваться в качестве корневой файловой системы.

Ext2, Ext3, Ext4 или Extended Filesystem – это стандартная файловая система для Linux. Она была разработана еще для Minix. Она самая стабильная из всех существующих. Её кодовая база изменяется очень редко. Кроме того, данная файловая система содержит больше всего функций. Версия ext2 была разработана уже именно для Linux и получила много улучшений.

JFS или Journaled File System была разработана в IBM для AIX UNIX и использовалась в качестве альтернативы для файловых систем ext. Сейчас она используется там, где необходима высокая стабильность и минимальное потребление ресурсов. При разработке файловой системы ставилась цель создать максимально эффективную файловую систему для многопроцессорных компьютеров. Также как и ext, это журналируемая файловая система, но в журнале хранятся только метаданные, что может привести к использованию старых версий файлов после сбоев.

XFS - журналируемая файловая система, однако в отличие от ext, в журнал записываются только изменения метаданных. Она используется по умолчанию в дистрибутивах на основе Red Hat. Из недостатков - это невозможность уменьшения размера, сложность восстановления данных и риск потери файлов при записи, если будет неожиданное отключение питания, поскольку большинство данных находится в памяти.

**4.** Команда `findmnt` используется для поиска примонтированных файловых систем. Она используется для поиска монтированных устройств, а также может монтировать или размонтировать их при необходимости.

Для просмотра всех примонтированных файловых систем использовать команду: `findmnt --all`

**5.** Каждый процесс в Linux имеет свой идентификатор, называемый PID. Перед тем, как выполнить остановку процесса, нужно определить его PID. Для этого воспользуемся командами `ps` и `grep`. Команда `ps` предназначена для вывода списка активных процессов в системе и информации о них. Команда

grep запускается одновременно с ps (в канале) и будет выполнять поиск по результатам команды ps.

Есть еще один более простой способ узнать PID процесса — это команда `pidof`, которая принимает в качестве параметра название процесса и выводит его PID.

Когда известен PID процесса, мы можем убить его командой `kill`. Команда `kill` принимает в качестве параметра PID процесса.

Вообще команда `kill` предназначена для отправки сигнала процессу. По умолчанию, если мы не указываем какой сигнал посылать, посылается сигнал `SIGTERM` (от слова *termination* — завершение). `SIGTERM` указывает процессу на то, что необходимо завершиться. Каждый сигнал имеет свой номер. `SIGTERM` имеет номер 15. Список всех сигналов (и их номеров), которые может послать команда `kill`, можно вывести, выполнив `kill -l`.

Сигнал `SIGTERM` может и не остановить процесс (например, при перехвате или блокировке сигнала), `SIGKILL` же выполняет уничтожение процесса всегда, так как его нельзя перехватить или проигнорировать.

Команда `killall` в Linux предназначена для «убийства» всех процессов, имеющих одно и то же имя. Это удобно, так как нам не нужно знать PID процесса.

**Вывод:** таким образом, мы приобрели практические навыки установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.